# 习题二

151250104, 卢以宁, kiwiloveskiwis@gmail.com

2017 年 4 月 6 日

## 1 [10pts] Lagrange Multiplier Methods

请通过拉格朗日乘子法 (可参见教材附录 B.1) 证明《机器学习》教材中式 (3.36) 与式 (3.37) 等价。即下面公式(1.1)与(1.2)等价。

$$\begin{aligned} \min_{\mathbf{w}} \quad & -\mathbf{w}^{\mathrm{T}}\mathbf{S}_b\mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w} = 1 \end{aligned} \tag{1.1}$$

$$\mathbf{S}_b\mathbf{w} = \lambda\mathbf{S}_w\mathbf{w} \tag{1.2}$$

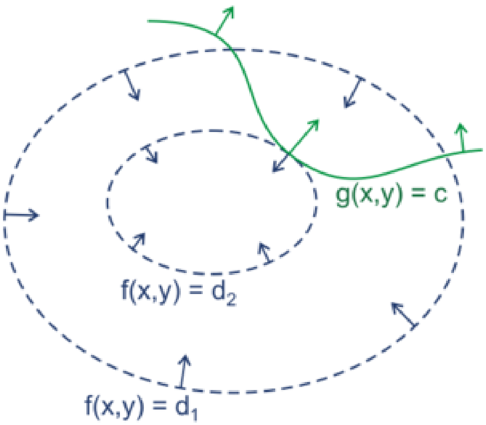


图 1: Contours of **g** and **f**

**Proof.** Denote the function we wish to minimize as $f(\mathbf{w}) = -\mathbf{w}^{\mathrm{T}}\mathbf{S}_b\mathbf{w}$, and the constraint function as $g(\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w} - 1$. Then, we should find a stationary point where $f(\mathbf{w})$ doesn't change along the contours[1]of $g(\mathbf{w}) = 0$ (Otherwise, then we can follow the direction where $\nabla_{\mathbf{w}} f < 0$ and get a smaller value ). In this case, the contour lines of **g** and **f** must be parallel, which indicates that the derivatives of **g** and **f** are also parallel [2]. Therefore:

$$\nabla_{\mathbf{w}} f + \lambda \nabla_{\mathbf{w}} g = 0 \tag{1.3}$$

Since

$$\nabla_{\mathbf{w}} f = -\frac{\partial}{\partial \mathbf{w}}\mathbf{w}^{\mathrm{T}}\mathbf{S}_b\mathbf{w} = -(\mathbf{S}_b + \mathbf{S}_b^T)\mathbf{w} = -2\mathbf{S}_b\mathbf{w} \tag{1.4}$$

$$\lambda \nabla_{\mathbf{w}} g = \lambda \frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^{\mathrm{T}} \mathbf{S}_w \mathbf{w} - 1) = \lambda (\mathbf{S}_w + \mathbf{S}_w^T) \mathbf{w} = 2\lambda \mathbf{S}_w \mathbf{w} \tag{1.5}$$

Combine them together, we finally get

$$\mathbf{S}_b \mathbf{w} = \lambda \mathbf{S}_w \mathbf{w} \tag{1.6}$$

□

## 2 [20pts] Multi-Class Logistic Regression

教材的章节 3.3 介绍了对数几率回归解决二分类问题的具体做法。假定现在的任务不再是二分类问题，而是多分类问题，其中 $y \in \{1, 2 \ldots, K\}$。请将对数几率回归算法拓展到该多分类问题。

(1) **[10pts]** 给出该对率回归模型的“对数似然”(log-likelihood);

(2) **[10pts]** 计算出该“对数似然”的梯度。

提示 1：假设该多分类问题满足如下 $K-1$ 个对数几率,

$$\begin{aligned}
\ln \frac{p(y=1|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_1^{\mathrm{T}}\mathbf{x} + b_1 \\
\ln \frac{p(y=2|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_2^{\mathrm{T}}\mathbf{x} + b_2 \\
&\cdots \\
\ln \frac{p(y=K-1|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_{K-1}^{\mathrm{T}}\mathbf{x} + b_{K-1}
\end{aligned}$$

提示 2：定义指示函数 $\mathbb{I}(\cdot)$,

$$\mathbb{I}(y=j) = \begin{cases} 1 & \text{若 } y \text{ 等于 } j \\ 0 & \text{若 } y \text{ 不等于 } j \end{cases}$$

**Solution.**

(1) We can run $K-1$ binary logistic regression model, where the $K$th outcome is chosen as the Pivot (Just as the Hint suggests). Therefore:

$$\begin{aligned}
\Pr(y=1|\mathbf{x}) &= \Pr(y=K|\mathbf{x}) e^{\mathbf{w}_1^{\mathrm{T}}\mathbf{x}+b_1} \\
\Pr(y=2|\mathbf{x}) &= \Pr(y=K|\mathbf{x}) e^{\mathbf{w}_2^{\mathrm{T}}\mathbf{x}+b_2} \\
&\cdots\cdots \\
\Pr(y=K-1|\mathbf{x}) &= \Pr(y=K|\mathbf{x}) e^{\mathbf{w}_{K-1}^{\mathrm{T}}\mathbf{x}+b_{K-1}}
\end{aligned} \tag{2.1}$$

[1]Image Reference: https://cuhkmath.wordpress.com/2010/10/12/understanding-lagrange-multipliers/

[2]Wikipedia - Lagrange multiplier

Since the sum of all above possibilities equals to 1, we get:

$$\begin{aligned} \Pr(y=1|\mathbf{x}) &= \frac{e^{\mathbf{w}_1^{\mathrm{T}}\mathbf{x}+b_1}}{1+\sum_{k=1}^{K-1} e^{\mathbf{w}_k^{\mathrm{T}}\mathbf{x}+b_k}} \\ \Pr(y=2|\mathbf{x}) &= \frac{e^{\mathbf{w}_2^{\mathrm{T}}\mathbf{x}+b_2}}{1+\sum_{k=1}^{K-1} e^{\mathbf{w}_k^{\mathrm{T}}\mathbf{x}+b_k}} \\ &\cdots\cdots \\ \Pr(y=K-1|\mathbf{x}) &= \frac{e^{\mathbf{w}_{K-1}^{\mathrm{T}}\mathbf{x}+b_{K-1}}}{1+\sum_{k=1}^{K-1} e^{\mathbf{w}_k^{\mathrm{T}}\mathbf{x}+b_k}} \\ \Pr(y=K|\mathbf{x}) &= \frac{1}{1+\sum_{k=1}^{K-1} e^{\mathbf{w}_k^{\mathrm{T}}\mathbf{x}+b_k}} \end{aligned} \tag{2.2}$$

Therefore, let $\boldsymbol{\beta}_\mathbf{i} = (\mathbf{w}_\mathbf{i}; b_i)$ and $\hat{\mathbf{x}}_\mathbf{i} = (\mathbf{x}_\mathbf{i}; 1)$, given dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ the log-likelihood should be:

$$\ell(\boldsymbol{\beta}) = \sum_{t=1}^{m} \Big( \sum_{k=1}^{K-1} \mathbb{I}(y_t = k) {\boldsymbol{\beta}_\mathbf{k}}^T \hat{\mathbf{x}}_\mathbf{t} - \ln(1 + \sum_{k=1}^{K-1} e^{{\boldsymbol{\beta}_\mathbf{k}}^T \hat{\mathbf{x}}_\mathbf{t}}) \Big) \tag{2.3}$$

(2) The derivative is

$$\frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_i} = \sum_{t=1}^{m} \Big( \mathbb{I}(y_t = i) \hat{\mathbf{x}}_\mathbf{t} - \mathbb{I}(y_t \neq K) \frac{\hat{\mathbf{x}}_\mathbf{t} \cdot e^{{\boldsymbol{\beta}_\mathbf{i}}^T \hat{\mathbf{x}}_\mathbf{t}}}{1 + \sum_{k=1}^{K-1} e^{{\boldsymbol{\beta}_\mathbf{k}}^T \hat{\mathbf{x}}_\mathbf{t}}} \Big) \tag{2.4}$$

# 3 [35pts] Logistic Regression in Practice

对数几率回归 (Logistic Regression, 简称 LR) 是实际应用中非常常用的分类学习算法。

(1) **[30pts]** 请编程实现二分类的 LR, 要求采用牛顿法进行优化求解, 其更新公式可参考《机器学习》教材公式 (3.29)。详细编程题指南请参见链接：http://lamda.nju.edu.cn/ml2017/PS2/ML2_programming.html

(2) **[5pts]** 请简要谈谈你对本次编程实践的感想 (如过程中遇到哪些障碍以及如何解决, 对编程实践作业的建议与意见等)。

**Solution.** (2) The problem of Overflow and Underflow happens a lot. Take the sigmoid function as an example, when -x is sufficiently large, `np.exp(-x)` would raise `Overflow Exception`. To avoid this, I re-write the function in Python as follows:

```
def sigmoid(x):
    max_elem = max(-x)
    try:
```

```
    ans = np.exp(-(np.log(np.exp(0 - max_elem) + np.exp(- x - max_elem)) + max_elem ) )
except Exception as e:
    ans = 0
return res
```

The principle behind is:

$$\log\left(e^a + e^b\right) = \log\left(e^{a-m} + e^{b-m}\right) + m \tag{3.1}$$

Then only underflow would happen. In this case, since the value is sufficiently low, we dismiss the exception and set `ans` to 0.

Another problem evolves the `SingularMatrix Exception` when running `np.linalg.inv(hess)`. Therefore, we could catch the exception and try to determine whether Hessian matrix is too small, if it does, we alternate to gradient descent.

```
try:
    inv = np.linalg.inv(hess)
    beta -= np.matmul(inv, grad(X, beta, y))
except Exception as e:
    if(np.max(hess) < np.exp(-100)):
        break
    else:
        beta = beta_save - grad(X, beta, y)
```

The third problem is about the initial value of $w$. When set to all ones, the training algorithm would never converge. This problem was finally found and fixed by setting $w$ to all zeros, after debugging in the dormitory for a whole spring morning ;_;.

# 4 [35pts] Linear Regression with Regularization Term

给定数据集 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_m, y_m)\}$, 其中 $\mathbf{x}_i = (x_{i1}; x_{i2}; \cdots; x_{id}) \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, 当我们采用线性回归模型求解时, 实际上是在求解下述优化问题：

$$\hat{\mathbf{w}}^*_{\mathbf{LS}} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2, \tag{4.1}$$

其中, $\mathbf{y} = [y_1, \cdots, y_m]^{\mathrm{T}} \in \mathbb{R}^m, \mathbf{X} = [\mathbf{x}_1^{\mathrm{T}}; \mathbf{x}_2^{\mathrm{T}}; \cdots; \mathbf{x}_m^{\mathrm{T}}] \in \mathbb{R}^{m\times d}$, 下面的问题中, 为简化求解过程, 我们暂不考虑线性回归中的截距 (intercept)。

在实际问题中, 我们常常不会直接利用线性回归对数据进行拟合, 这是因为当样本特征很多, 而样本数相对较少时, 直接线性回归很容易陷入过拟合。为缓解过拟合问题, 常对公式(4.1)引入正则化项, 通常形式如下：

$$\hat{\mathbf{w}}^*_{\mathbf{reg}} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\Omega(\mathbf{w}), \tag{4.2}$$

其中, $\lambda > 0$ 为正则化参数, $\Omega(\mathbf{w})$ 是正则化项, 根据模型偏好选择不同的 $\Omega$。

下面, 假设样本特征矩阵 $\mathbf{X}$ 满足列正交性质, 即 $\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{I}$, 其中 $\mathbf{I} \in \mathbb{R}^{d\times d}$ 是单位矩阵, 请回答下面的问题 (需要给出详细的求解过程)：

(1) **[5pts]** 考虑线性回归问题, 即对应于公式(4.1), 请给出最优解 $\hat{\mathbf{w}}^*_{\mathbf{LS}}$ 的闭式解表达式;

(2) **[10pts]** 考虑岭回归 (ridge regression)问题, 即对应于公式(4.2)中 $\Omega(\mathbf{w}) = \|\mathbf{w}\|_2^2 = \sum_{i=1}^d w_i^2$ 时, 请给出最优解 $\hat{\mathbf{w}}^*_{\mathbf{Ridge}}$ 的闭式解表达式;

(3) **[10pts]** 考虑LASSO问题, 即对应于公式(4.2)中 $\Omega(\mathbf{w}) = \|\mathbf{w}\|_1 = \sum_{i=1}^d |w_i|$ 时, 请给出最优解 $\hat{\mathbf{w}}^*_{\mathbf{LASSO}}$ 的闭式解表达式;

(4) **[10pts]** 考虑 $\ell_0$-范数正则化问题,

$$\hat{\mathbf{w}}^*_{\ell_0} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_0, \tag{4.3}$$

其中, $\|\mathbf{w}\|_0 = \sum_{i=1}^d \mathbb{I}[w_i \neq 0]$, 即 $\|\mathbf{w}\|_0$ 表示 $\mathbf{w}$ 中非零项的个数。通常来说, 上述问题是 NP-Hard 问题, 且是非凸问题, 很难进行有效地优化得到最优解。实际上, 问题 (3) 中的 LASSO 可以视为是近些年研究者求解 $\ell_0$-范数正则化的凸松弛问题。

但当假设样本特征矩阵 $\mathbf{X}$ 满足列正交性质, 即 $\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{I}$ 时, $\ell_0$-范数正则化问题存在闭式解。请给出最优解 $\hat{\mathbf{w}}^*_{\ell_0}$ 的闭式解表达式, 并简要说明若去除列正交性质假设后, 为什么问题会变得非常困难？

**Solution.**

(1) Let

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 = \frac{1}{2}(\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y}) \\ &= \frac{1}{2}((\mathbf{X}\mathbf{w})^{\mathbf{T}}\mathbf{X}\mathbf{w} - (\mathbf{X}\mathbf{w})^{\mathbf{T}}\mathbf{y} - \mathbf{y}^{\mathbf{T}}(\mathbf{X}\mathbf{w}) + \mathbf{y}^{\mathbf{T}}\mathbf{y}) \end{aligned} \tag{4.4}$$

Since

$$\frac{\partial J}{\partial \mathbf{w}} = \mathbf{X}^{\mathbf{T}}\mathbf{X}\mathbf{w} - \mathbf{X}^{\mathbf{T}}\mathbf{y} \tag{4.5}$$

Let (4.5) $= 0$, since $\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{I}$, we have

$$\mathbf{w} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\mathbf{y} = \mathbf{X}^{\mathbf{T}}\mathbf{y} \tag{4.6}$$

(2) Let

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\lambda\|_2^2 + \|\mathbf{w}\|_2^2 \\ &= \frac{1}{2}((\mathbf{X}\mathbf{w})^{\mathbf{T}}\mathbf{X}\mathbf{w} - \mathbf{2}(\mathbf{X}\mathbf{w})^{\mathbf{T}}\mathbf{y} + \mathbf{y}^{\mathbf{T}}\mathbf{y}) + \lambda\mathbf{w}^{\mathbf{T}}\mathbf{w} \end{aligned} \tag{4.7}$$

Since

$$\frac{\partial J}{\partial \mathbf{w}} = \mathbf{X}^{\mathbf{T}}\mathbf{X}\mathbf{w} - \mathbf{X}^{\mathbf{T}}\mathbf{y} + \lambda\mathbf{w} \tag{4.8}$$

Let (4.8) $= 0$, we have

$$\mathbf{w} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda\mathbf{I_d})^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\mathbf{y} = \frac{\mathbf{1}}{\mathbf{1} + \lambda}\mathbf{X}^{\mathbf{T}}\mathbf{y} \tag{4.9}$$

(3) Let $\hat{\mathbf{w}}^{\mathrm{LS}}$ denote the solution to (4.1)($i.e.$ $\hat{\mathbf{w}}^{\mathrm{LS}} = \mathbf{X^T y}$), we have:

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{2}\|\mathbf{y}-\mathbf{Xw}\|_2^2 + \lambda\|\mathbf{w}\|_1 \\ &= \frac{1}{2}(\mathbf{w^T X^T X w} - \mathbf{2(Xw)^T y} + \mathbf{y^T y}) + \lambda\|\mathbf{w}\|_\mathbf{1} \\ &= \frac{1}{2}(\sum_{\mathbf{i=1}}^{\mathbf{d}} \mathbf{w_i^2} - \mathbf{2wX^T y} + \mathbf{y^T y}) + \lambda\|\mathbf{w}\|_\mathbf{1} \\ &= \frac{1}{2}(\sum_{\mathbf{i=1}}^{\mathbf{d}} (\mathbf{w_i^2} - \mathbf{2w_i}\widehat{\mathbf{w}}_\mathbf{i}^{\mathrm{LS}}) + \mathbf{y^T y}) + \lambda\|\mathbf{w}\|_\mathbf{1} \end{aligned} \tag{4.10}$$

Since $\mathbf{y^T y}$ is irrelevant to $\mathbf{w}$ we have it discarded. Therefore:

$$\min_{\mathbf{w}} J(\mathbf{w}) = \min_{\mathbf{w}} \sum_{i=1}^{d} (\frac{1}{2}\mathbf{w}_i^2 - \mathbf{w}_i\hat{\mathbf{w}}_i^{\mathrm{LS}} + \lambda|\mathbf{w_i}|) \tag{4.11}$$

Where $\mathbf{w}_i$ is independent to $\mathbf{w}_j (i \neq j)$, and we can minimize the whole $J(\mathbf{w})$ by finding $\mathbf{w}_k (k = 1, 2, 3..., d)$ one by one, $i.e.$ for every $k \in [1, d]$, we need to find

$$\min_{\mathbf{w}_k} J(\mathbf{w}_k) = \min_{\mathbf{w}_k} (\frac{1}{2}\mathbf{w}_k^2 - \mathbf{w}_k\hat{\mathbf{w}}_k^{\mathrm{LS}} + \lambda|\mathbf{w_k}|) \tag{4.12}$$

Since the value of $\frac{1}{2}\mathbf{w}_k^2$ and $\lambda|\mathbf{w_k}|$ is independent to the sign, if $\hat{\mathbf{w}}_k^{\mathrm{LS}} > 0$ then $\mathbf{w}_k$ must be $\geq 0$. If $\hat{\mathbf{w}}_k^{\mathrm{LS}} \leq 0$ then $\mathbf{w}_k \leq 0$. Then:

$$J(\mathbf{w}_k) = \begin{cases} \frac{1}{2}\mathbf{w}_k^2 - \mathbf{w}_k\hat{\mathbf{w}}_k^{\mathrm{LS}} + \lambda\mathbf{w}_k & , \hat{\mathbf{w}}_k^{\mathrm{LS}} > 0 \\ \frac{1}{2}\mathbf{w}_k^2 - \mathbf{w}_k\hat{\mathbf{w}}_k^{\mathrm{LS}} - \lambda\mathbf{w}_k & , \hat{\mathbf{w}}_k^{\mathrm{LS}} \leq 0 \end{cases} \tag{4.13}$$

In either case, we have:

$$\frac{\partial J(\mathbf{w}_k)}{\partial \mathbf{w_k}} = \mathbf{w}_k - \hat{\mathbf{w}}_k^{\mathrm{LS}} + \mathrm{sign}(\hat{\mathbf{w}}_k^{\mathrm{LS}}) \cdot \lambda \tag{4.14}$$

Therefore, the closed-form solution is given by:[3]

$$\hat{\mathbf{w}}_k^{\mathrm{lasso}} = \begin{cases} \hat{\mathbf{w}}_k^{\mathrm{LS}} - \mathrm{sign}(\hat{\mathbf{w}}_k^{\mathrm{LS}}) \cdot \lambda & , \lambda < |\hat{\mathbf{w}}_k^{\mathrm{LS}}| \\ 0 & , \lambda > |\hat{\mathbf{w}}_k^{\mathrm{LS}}| \end{cases} \tag{4.15}$$

(4) Let

$$J(\mathbf{w}) = \frac{1}{2}((\sum_{\mathbf{i=1}}^{\mathbf{d}} (\mathbf{w_i^2} - \mathbf{2w_i}\widehat{\mathbf{w}}_\mathbf{i}^{\mathrm{LS}}) + \mathbf{y^T y}) + \lambda\|\mathbf{w}\|_\mathbf{0} \tag{4.16}$$

Similar to (4.13), we have:

$$J(\mathbf{w}_k) = \begin{cases} 0 & , \mathbf{w}_k = 0 \\ \frac{1}{2}\mathbf{w}_k^2 - \mathbf{w}_k\hat{\mathbf{w}}_k^{\mathrm{LS}} + \lambda & , \mathbf{w}_k \neq 0 \end{cases} \tag{4.17}$$

Therefore, from basic principles of the quadratic equation, we have

$$\hat{\mathbf{w}}_k^{\ell_0} = \begin{cases} \hat{\mathbf{w}}_k^{\mathrm{LS}} \pm \sqrt{(\hat{\mathbf{w}}_k^{\mathrm{LS}})^2 - 2\lambda} & , (\hat{\mathbf{w}}_k^{\mathrm{LS}})^2 - 2\lambda > 0 \\ 0 & , (\hat{\mathbf{w}}_k^{\mathrm{LS}})^2 - 2\lambda \leq 0 \end{cases} \tag{4.18}$$

if $\mathbf{X}$ is non-orthonormal, since the $L0$-penalty makes the solution non-linear, rendering the minimization a quadratic programming problem, which is NP-hard in general. [4,5]

[3] C Leng. A note on the Lasso in Model Selection Statistica Sinica 16(2006), 1273-1284

[4] Cedric E. Ginestet, Regularization: Ridge Regression and Lasso, Boston University, MA 575 Linear Models, Week 14, Lecture 2

[5] SA Vavasis, Quadratic programming is in NP, 1990-02, Cornell University